容器和 Kubernetes 安全防护分层方法

确保容器从构建、部署到运行的安全

目录

简介 2

全面容器和 Kubernetes 安全防护：层级和生命周期 2

确保应用安全无虞 4

部署：管理部署的配置、安全和合规性 8

保护正在运行的应用 11

借助强大的生态系统扩展安全性 15

结论 15

红帽官方微博　红帽官方微信

简介

容器具有广泛的吸引力，因为容器能够将应用及其依赖项封装到单个镜像中，而且这类镜像可用于开发、测试和生产。容器能让您轻松地在多个环境和部署目标（如物理服务器、虚拟机（VM）和私有云或公共云）间保持一致性。借助容器，团队可以更加轻松地开发和管理应用，进而创造业务敏捷性。

- **应用：**借助容器，开发人员可将应用及其依赖项作为一个单元来处理，以简化相关的构建和应用工作。只需几分秒即可部署容器。在容器化环境中，软件构建流程是指生命周期中将应用代码与所需的运行时库进行整合的阶段。
- **基础架构：**容器代表共享 Linux® 操作系统内核上的沙盒化应用进程。和虚拟机相比，它们更紧凑、更轻、更简单，并且可以从本地到公共云平台跨不同环境移植。

Kubernetes 是企业首选的容器编排平台。现在许多企业都在容器上运行重要服务，确保容器安全从未如此重要。本文将介绍容器化应用安全防护的关键要素。

全面容器和 Kubernetes 安全防护：层级和生命周期

确保容器的安全与确保所有运行中的 Linux 进程的安全很类似。在部署和运行容器之前，您要全面考虑解决方案堆栈的各个层级的安全性。您还需要全面考虑应用和容器整个生命周期的安全性。重要的是，必须确保整个 IT 生命周期的持续安全防护，来应对新出现的威胁和解决方案。图 1 阐述了全面的容器安全防护方法。

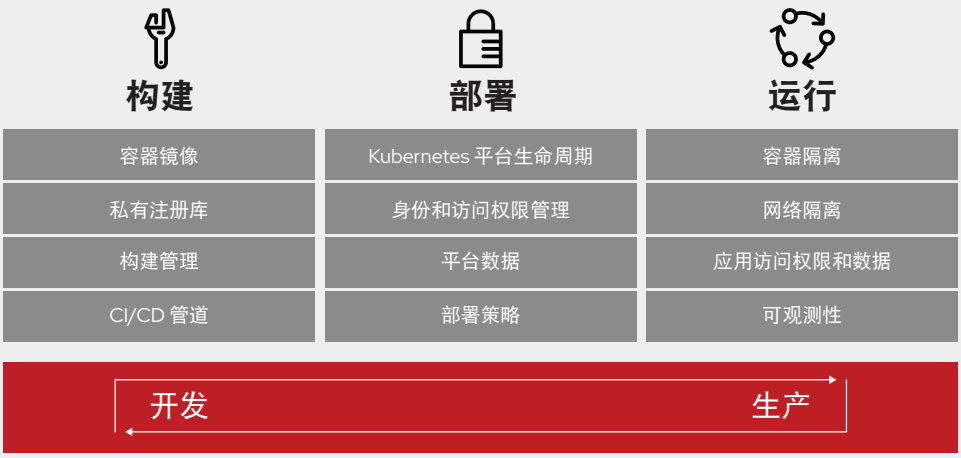

图 1. 全面的容器安全防护包括确保构建、部署和运行时的安全，并在适当的时候用专业工具扩展安全。

借助容器，开发人员可将应用及其依赖项作为一个单元来进行处理，以简化相关的构建和应用工作。容器还可在共享主机上实现多租户应用部署，从而轻松地实现对服务器的充分利用。您可以轻松地在单个主机上部署多个应用，并根据需要启用和关闭单个容器。您也不需要在各个 VM 上分别安装虚拟机监控程序以管理虚拟客户机操作系统，这一点有别于传统虚拟化。容器能让应用进程（而非硬件）实现虚拟化。

当然，应用很少会以单个容器的形式来交付。即使是非常简单的应用，通常也会由前端、后端和数据库构成。在容器中部署基于微服务的现代化应用意味着部署多个容器——有时在同一个主机上，有时分布在多个主机或节点上（如图 2 所示）。

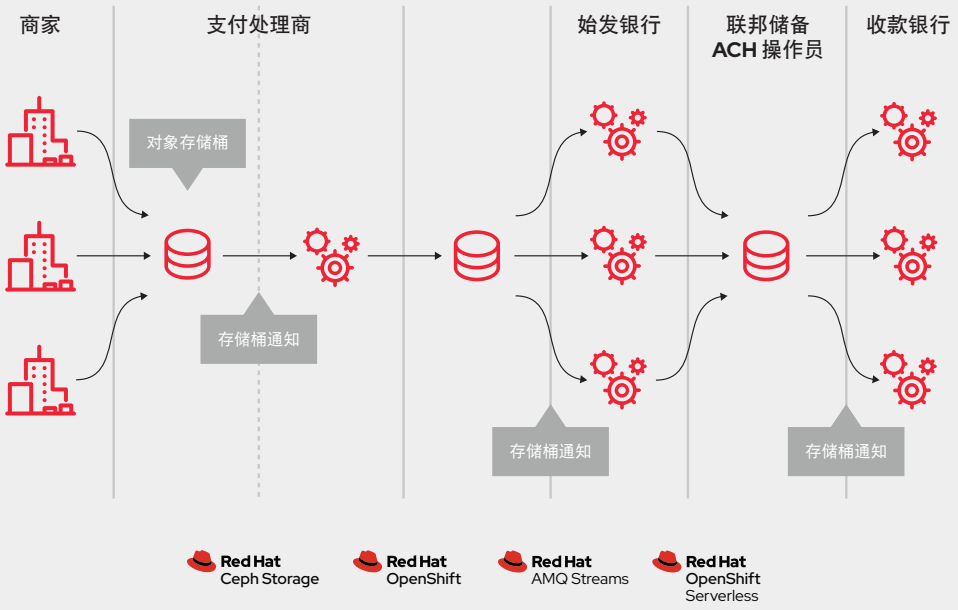

图 2. 容器环境可能很复杂，且分布在多个节点上。

在管理大规模的容器部署时，您需要考虑：

- 应将哪个容器部署到哪个主机？
- 要为哪个主机分配更多容量？
- 哪些容器需要相互访问，以及它们要如何发现彼此？
- 如何控制对共享资源（如网络和存储）的访问和管理？
- 如何监控容器的健康状况？
- 如何自动扩展应用容量以满足需求？
- 如何在满足安全要求的情况下启用开发人员自助服务？

您可以构建自己的容器管理环境，这需要花费时间去集成和管理各个组件。或者您可以使用内置管理和安全功能来部署容器平台。这种方法可以让您的团队将精力集中于构建能够创造业务价值的应用上，而非重复发明基础架构。

红帽® OpenShift® 容器平台提供一致的混合云企业 Kubernetes 平台，用于构建和扩展容器化应用。在整个企业内使用 Kubernetes 需要帮助您确保应用安全无虞的额外安全功能、让您管理容器部署安全的自动化策略，以及保护容器运行时的功能。

确保应用安全无虞

确保应用安全无虞对于云原生部署至关重要。确保容器化应用安全需要：

1. 使用可信的容器内容。
2. 使用企业容器镜像仓库。
3. 控制和自动化容器构建。
4. 将安全性整合到应用管道。

1. 使用可信的容器内容

管理安全性时，重要的是保护容器中所含的内容。一段时间以来，应用和基础架构一直都是由各种即用型组件组合而成的。而且，其中的很多组件都是开源数据包，如 Linux 操作系统、Apache Web 服务器、红帽 JBoss® 企业应用平台、PostgreSQL 和 Node.js。这些数据包的容器化版本已准备就绪，您不必再自行构建。但是，在从外部来源下载任意代码时，您需要知晓这些数据包的原始来源、构建者以及其中是否含有任何恶意代码。请思考几个问题：

- 容器内容是否会危及我的基础架构？
- 应用层是否存在已知的漏洞？
- 容器中的运行时和操作系统层是否处于最新状态？
- 容器将多久更新一次？当容器更新时，我将如何知晓这一情况？

多年来，红帽一直通过红帽企业 Linux 和我们的产品组合来封装和交付可信的 Linux 内容。现在，红帽正通过封装成 Linux 容器的方式来交付同样可信的内容，红帽通用基础镜像推出后，无论在哪里运行符合开放容器计划（OCI）的 Linux 容器，您都可以充分利用红帽容器镜像更高的可靠性、安全性和性能。这意味着您可以在红帽通用基础镜像上构建容器化应用，将其推送到您所选的容器镜像仓库并进行共享。

红帽还通过红帽生态系统目录针对各种语言运行时、中间件、数据库等提供大量经过认证的镜像和操作器。不管是裸机、 VM 还是云端，只要能运行红帽企业 Linux，就可以运行经过红帽认证的容器和操作器，并由红帽和我们的合作伙伴提供支持。

红帽持续监控所提供镜像的健康状态。容器健康指数揭示了各个容器镜像的“等级”，从而详细指明应该如何管辖、使用和评估容器镜像，以满足生产系统的需求。在对容器进行评级时，所考虑的部分因素是未应用于容器内所有组件的安全勘误的已存在时长和所造成的影响，这样能得出一个安全专家和非专家人士都能理解的容器安全性总体评级。

当红帽发布安全性更新时，如针对 CVE-2019-5736、MDS CVE-2019-11091，或 VHOST-NET CVE-2019-14835 运行修复，我们还会重构我们的容器镜像并将其推送到我们的公共注册表。红帽安全公告会向您发出提醒，告知您我们在认证容器镜像中新发现的所有问题，并指引您找到更新后的镜像，以便您转而更新使用该镜像的所有应用。

有时您可能需要使用红帽并未提供的内容。我们建议您使用漏洞数据库会不断更新的容器扫描工具，以确保您在使用其他来源的容器镜像时始终能获得有关已知漏洞的最新信息。由于已知漏洞列表会不断变更，所以您在首次下载容器镜像时需要检查其所含的内容，并持续跟踪所有已获批和已部署镜像的漏洞状态，就像红帽跟踪红帽容器镜像那样。

2. 使用企业容器镜像仓库，更安全地访问容器镜像

当然，您的团队会构建内容基于所下载公共容器镜像的容器。您需要按照您在管理其他类型的二进制文件时所用的方式，来管理对已下载的容器镜像和内部构建镜像的访问权限及其应用方式。支持容器镜像存储的私有注册表有很多。我们建议，您选择的私有注册表应能帮助您自动实施与使用注册表中所存储容器镜像相关的策略。

红帽 OpenShift 包含提供基础功能以管理容器镜像的私有注册表。红帽 OpenShift 注册表提供了基于角色的访问权限控制（RBAC），允许您管理哪些人员可以提取和推送特定的容器镜像。红帽 OpenShift 还支持与您可能已在使用的其他私有注册表进行整合，如 JFrog 的 Artifactory 以及 Sonatype Nexus。

红帽 Quay 可以作为一个单独使用的企业注册表。红帽 Quay 提供许多额外的企业功能，如地理位置复制和构建镜像触发。

Clair 项目是一个开源引擎，可为红帽 Quay 安全扫描程序提供支持，从而检测红帽 Quay 内所有镜像的漏洞。红帽 OpenShift 容器安全操作器与红帽 Quay 集成在一起，为您提供在 OpenShift 控制台中部署镜像的已知漏洞的集群视图。

3. 控制和自动化容器镜像构建

如何管理这个构建流程，是确保软件堆栈安全性的关键所在。如能遵循“一次构建，随心部署”的理念，即可确保构建流程的产物就是要在生产环境中部署的项目。另外，保持容器的不可变性也十分重要。换句话说，就是不要为正在运行的容器安装补丁，而应重新构建并重新部署这些容器。

红帽 OpenShift 提供了多种功能，可用于基于外部事件的自动化构建，从而提高自定义镜像的安全性。

- 红帽 Quay 触发器提供了一种机制，用于从外部事件（如 GitHub 推送、BitBucket 推送、GitLab 推送或 webhook）生成 Dockerfile 的存储库构建。
- 源至镜像（S2I）是一个用于组合源代码和基本镜像的开源框架。（图 3）S2I 能让开发和运维团队轻松地在能够重现的构建环境中开展协作。当开发人员在 S2I 下使用 Git 提交代码时，红帽 OpenShift 可以：
  - 触发（通过代码存储库中的 webhook 或是其他的一些自动 CI 流程）从可用的工件（包括 S2I 基本镜像）和新提交的代码中自动组装新的镜像。
  - 自动部署新构建的镜像，以进行测试。
  - 将经过测试的镜像置于生产状态，并通过持续集成和持续部署（CI/CD）流程自动部署新的镜像。

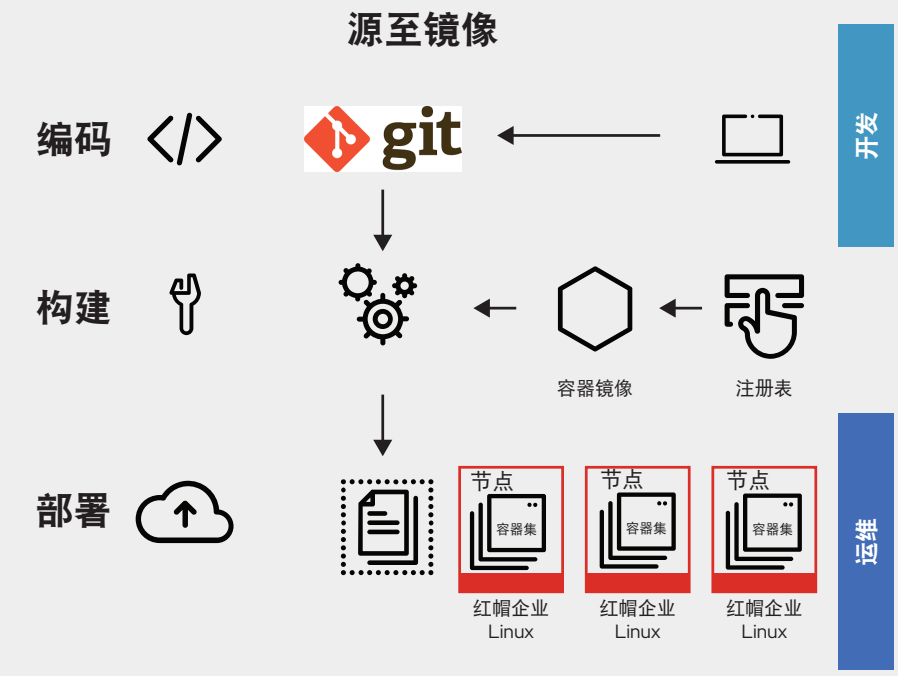

图 3. S2I 是工具包和工作流程，用于从源代码中构建可重现、可直接运行的容器镜像。

- 红帽 OpenShift 镜像流可用于观察部署在集群中的外部镜像的变化。镜像流与红帽 OpenShift 中可用的所有原生资源（如构件或部署、工作、复制控制器或副本集）协同运作。通过观察镜像流，当新的镜像被添加或修改时，构件和部署可以收到通知，并分别通过自动启动构件或部署作出反应。

  例如，考虑用基础层、中间件层和应用层这三个容器镜像层构建的应用。红帽发现基础镜像存有问题，然后对该镜像进行重新构建并推送至红帽生态系统目录。启用镜像流后，红帽 OpenShift 便会检测到该镜像已被更改。对于依赖于该镜像的构件和已定义触发器的构件，红帽 OpenShift 将自动重新构建这个应用镜像，并引入已修复的基础镜像。

  构建完成后，该已更新的自定义镜像会被推送到红帽 OpenShift 的内部注册表。红帽 OpenShift 会即刻检测出其内部注册表中的镜像发生了变化，并自动为已定义触发器的应用部署更新后的镜像，以确保生产环境中运行的代码与最新的更新后镜像始终保持一致。所有这些功能通过协同合作，确保了 CI/CD 流程和管道的安全性。

4. 将安全性整合到应用管道

红帽 OpenShift 包含可用于实现 CI 和 Tekton 的集成式 Jenkins 实例。Tekton 是新一代 Kubernetes CI/CD 管道，适用于容器（包括无服务器）。红帽 OpenShift 还包含丰富多样的 RESTful API，可供您用于整合自己的构建工具或 CI/CD 工具（包括私有镜像注册表）。

应用安全防护的最佳实践是将自动安全测试集成到您的管道，包括注册表、集成开发环境（IDE）和 CI/CD 工具。

注册表：容器镜像可以而且应该在您的私有容器镜像仓库中进行扫描。您可以使用集成 Clair 安全扫描程序的红帽 Quay，在发现漏洞时通知开发人员。OpenShift 容器安全操作器与红帽 Quay 集成在一起，为您提供 OpenShift 控制台中已部署镜像的已知漏洞的集群视图。另外，您可以在红帽生态系统目录中找到多个第三方认证的容器扫描解决方案。

IDE：红帽依赖分析集成开发环境（IDE）插件在代码首次被带入 IDE 时，为项目依赖项提供漏洞警告和修复建议。

CI/CD：可与 CI 集成，用于实时检查已知漏洞的扫描程序可为容器中的开源数据包编目、就任何已知漏洞向您发出通知并在先前所扫描数据包中发现新漏洞时向您提供最新信息。

另外，CI 流程还应包含相应策略，以标记通过安全扫描发现有问题的构件。这样，您的团队便可采取相应措施，尽早处理这些问题。

最后，我们建议您对定制容器进行签名，以确保它们不会在构建和部署过程中被篡改。

部署：管理部署的配置、安全和合规性

部署的有效安全防护包括确保 Kubernetes 平台的安全以及部署策略的自动化。红帽 OpenShift 包含以下开箱即用功能：

1. 平台配置和生命周期管理。
2. 身份和访问权限管理。
3. 确保平台数据和附加存储的安全。
4. 部署策略。

5. 平台配置和生命周期管理

于 2019 年夏季发布的云原生计算基金会（CNCF）Kubernetes 安全审计总结道，Kubernetes 最大的安全威胁是配置和强化 Kubernetes 组件的复杂性。红帽 OpenShift 通过使用 Kubernetes 操作器来应对这一挑战。

操作器是一种封装、部署和管理 Kubernetes 原生应用的方法。操作器作为一个自定义的控制器，可以借助管理应用所需的特定于应用的逻辑来扩展 Kubernetes 应用编程接口（API）。每个红帽 OpenShift 平台组件都包含在一个操作器中，为 OpenShift 提供自动化配置、监控和管理。单个操作器直接配置 API 服务器和软件定义网络（SDN）等组件，而集群版操作器则管理整个平台的多个操作器。操作器可让您实现集群管理自动化（从内核到堆栈中更高的服务），包括更新。

容器平台的另一大重要价值在于，它能为开发人员提供自助服务，从而使得开发团队能够更加轻松快速地交付基于已批准层级构建的应用。自助服务门户可为您的团队提供足够的控制权，以在确保安全性的情况下促进合作。操作器生命周期管理器（OLM）为红帽 OpenShift 集群用户提供了一个框架，用于寻找和使用操作器来部署启用其应用所需的服务。借助 OLM，用户可以安装、升级，并为可用的操作器分配基于角色的访问权限控制。

为了实现合规性，红帽 OpenShift 提供了合规性操作器，以实现平台合规性自动化，同时具备合规性框架所要求的技术控制。合规性操作器可让红帽 OpenShift 管理员描述集群所需的合规性状态，并为他们提供差距和补救方法的概述。合规性操作器评估所有平台层的合规性，包括运行集群的节点。文件完整性操作器还可以定期在集群节点上运行文件完整性检查。

6. 身份和访问权限管理

由于开发人员和管理员都有大量的 Kubernetes 功能可以使用，所以强大的身份管理和 RBAC 是容器平台的关键要素之一。Kubernetes API 是实现大规模自动化容器管理的关键所在。例如，API 用于启动和验证请求，包括配置和部署容器集和服务。

API 身份验证和授权是确保容器平台安全的关键所在。API 服务器是访问的中央点，应该接受最高等级的安全审查。红帽 OpenShift 控制平面包括通过集群身份验证操作器的内置身份验证。开发人员、管理员和服务账户可以获取 OAuth 访问令牌，以完成他们的 API 验证。作为管理员，您可以将您所选的身份提供商配置到集群，这样用户就可以在收到令牌之前进行身份验证。支持九个身份提供商，包括轻量级目录访问协议（LDAP）目录。

红帽 OpenShift 中默认启用精细 RBAC。RBAC 对象决定用户是否被允许在集群中执行给定的操作。集群管理员可以使用集群角色和绑定来控制对 OpenShift 集群和集群内项目的访问级别。

7. 确保平台数据的安全

红帽 OpenShift 默认强化 Kubernetes，以确保传输中的数据安全。还包括确保静止数据安全的选项。

红帽 OpenShift 通过以下方式保护传输中的平台数据：

- 所有相互通信的容器平台组件都通过 https 加密传输中的数据。
- 通过传输层安全（TLS）发送与控制平面的所有通信。
- 确保对 API 服务器的访问是基于 X.509 证书或基于令牌的。
- 使用项目配额来限制恶意令牌可能会导致的损害程度。
- 使用其证书颁发机构（CA）和证书配置 etcd。（在 Kubernetes 中，etcd 存储持久的主状态，而其他组件则监视 etcd 的变化，使自己进入指定状态。）
- 自动轮转平台证书。

红帽 OpenShift 通过以下方式保护静止的平台数据：

- 可选择对红帽企业 Linux CoreOS 磁盘和 etcd 数据存储进行加密，来获得额外的安全防护。
- 为红帽 OpenShift 提供联邦信息处理标准（FIPS）准备。FIPS 140-2 是美国政府用于批准加密模块的安全标准。红帽企业 Linux CoreOS 在 FIPS 模式下启动时，红帽 OpenShift 平台组件会调用红帽企业 Linux 加密模块。

容器对于无状态和有状态应用都大有帮助。红帽 OpenShift 支持临时和持久存储。保护附加存储是确保有状态服务安全无虞的关键所在。红帽 OpenShift 支持多种存储类型，包括网络文件系统（NFS）、Amazon Web Services（AWS）弹性块存储（EBS）、Google 计算引擎（GCE）持久磁盘、Azure 磁盘、iSCSI 和 Cinder。

此外，红帽 OpenShift 容器存储是与红帽 OpenShift 容器平台集成并优化的持久软件定义存储。OpenShift 容器存储为需要加密、复制和跨混合多云可用性等功能的云原生应用提供高度可扩展的持久存储。

- **持久卷（PV）**能以资源提供商支持的任意方式挂载到主机上。提供商将获得不同的功能，而且每个 PV 的访问模式都会被设置为该特定卷所支持的各种具体模式。例如，NFS 可以支持多个读/写客户端，但是特定的 NFS PV 可能会在服务器上导出为只读模式。每个 PV 都会获得自己的访问模式集，其中描述了相应 PV 的功能。例如，ReadWriteOnce、ReadOnlyMany 和 ReadWriteMany。

- 对于**共享存储**（例如，NFS、Ceph、 Gluster），有一个实用的技巧，那就是：将共享存储持久卷（PV）的组 ID（gid）作为注释在 PV 资源上进行注册。当容器集声明这个 PV 时，所注释的 gid 将会添加到该容器集的附加组中，并为该容器集提供访问共享存储中所含内容的权限。
- 对于**块存储**（例如，EBS、GCE 持久磁盘、iSCSI），容器平台可以使用 SELinux 功能来保护非特权容器集已挂载卷的根，以使已挂载卷归其关联容器所有，并且只有关联容器才能看到。

当然，您应充分利用所选存储解决方案中提供的各种安全功能。

8. 自动化基于策略的部署工作

高安全性包括可用于从安全角度来管理容器和集群部署的自动化策略。

- 基于策略的容器部署

红帽 OpenShift 集群可以配置为允许或不允许从特定的镜像仓库提取镜像。对于生产集群来说，最佳实践是只允许从您的私有注册库部署镜像。

红帽 OpenShift 的安全上下文约束（SCC）许可控制器插件定义了一组条件，容器集必须在这些条件下运行才能被系统接受。**安全上下文约束**让您默认放弃特权，这非常重要，而且仍是最为有效的一种方式。红帽 OpenShift 的安全上下文约束（SCC）确保在默认情况下，OpenShift 工作节点上没有特权容器运行。默认拒绝访问主机网络和主机进程 ID。

拥有所需权限的用户可以选择将默认 SCC 策略调整得更为宽容。

用于 Kubernetes 的红帽高级集群管理提供**高级应用生命周期管理**，利用集成到现有 CI/CD 管道和监管控制中的放置策略，应用开放标准来部署应用。

- 基于策略的多集群管理

部署多个集群可用于提供跨多个可用性区域的应用高可用性，或为跨多个云提供商（如 Amazon Web Services（AWS）、Google 云和 Microsoft Azure）的部署或迁移提供常规管理功能。在管理多个集群时，您的编排工具须能够跨不同的部署实例提供您所需的安全性。一如既往，配置、身份验证和授权仍是关键所在，另外，能够安全地将数据传输至应用（无论它们在何处运行）并实现跨集群管理应用策略的能力也非常重要。用于 Kubernetes 的红帽高级集群管理提供：

- **多集群生命周期管理**允许您大规模且可靠一致地创建、更新和销毁 Kubernetes 集群。
- **策略驱动型风险监管和合规性**根据行业公司标准，利用策略自动配置和维护安全控制的一致性。您还可以指定合规性策略，来应用于一个或多个受管集群。

保护正在运行的应用

除了基础架构之外，维护应用安全也至关重要。确保容器化应用安全需要：

1. 容器隔离。
2. 应用和网络隔离。
3. 确保应用的访问安全。
4. 可观测性。

9. 容器隔离

要充分利用容器封装和编排技术，运维团队需要为所运行的容器构建适用的环境。运维团队需要安装能在边界确保容器安全性的操作系统，从而避免主机内核出现容器逃逸情况，并确保容器不会相互影响。

容器就是相互隔离且存在资源约束的 Linux 进程，它们能让您在共享主机内核上运行沙盒化应用。您的容器安全性方案应与保障 Linux 上所有运行进程安全性时所用的方案保持一致。

NIST 特别出版物 800-190 建议使用容器优化的操作系统来获取额外的安全防护。作为红帽 OpenShift 的操作系统基础，红帽企业 Linux CoreOS 可最小化主机环境并根据容器对环境进行调优，从而减小攻击面。红帽企业 Linux CoreOS 只包含运行红帽 OpenShift 所需的包，其用户空间是只读的。该平台是与红帽 OpenShift 一起测试、版本化和传输的，并由集群管理。红帽企业 Linux CoreOS 的安装和更新是自动的，并且始终兼容集群。它还支持您所选择的基础架构，继承了大部分红帽企业 Linux 生态系统。

每一个在红帽 OpenShift 平台上运行的 Linux 容器都受到红帽 OpenShift 节点内置的强大红帽企业 Linux 安全功能保护。Linux 命名空间、SELinux、Cgroups、功能和安全计算模式（seccomp）是用于确保红帽企业 Linux 上所运行容器的安全性。

- Linux 命名空间能为实现容器隔离奠定基础。使用命名空间后，其中所含的进程看上去就像是拥有了自己的全局资源实例。命名空间可以实现抽象化，让您从容器内部感觉像是在自己的操作系统上运行一样。
- SELinux 可以提供额外的安全保护，以使容器相互隔离开并与主机隔离开。SELinux 允许管理员针对每一个用户、应用、进程和文件实施强制访问控制（MAC）。SELinux 就像一堵墙，当您设法突破（意外或故意）命名空间所形成的抽象化时，这堵墙会加以阻挡。SELinux 可减少容器运行时漏洞，配置良好的 SELinux 配置可以防止容器进程摆脱其控制。
- Cgroups（控制组）会限制、监管并隔离进程集合对资源的使用（如 CPU、内存、磁盘 I/O、网络）。它可以防止您的容器资源被同一主机上的其他容器影响。Cgroups 还可用于控制伪设备（一种常见的攻击向量）。
- Linux 功能可用于将特权锁定到容器中。功能是指可以单独启用或禁用的不同特权单元。借助功能，您可以执行发送原始互联网协议（IP）数据包或绑定至 1024 以下的端口等操作。运行容器时，您可在不影响绝大多数容器化应用的情况下删除多个功能。
- 最后，安全计算模式（seccomp）配置文件可与容器关联，限制可执行的系统调用。

10. 应用和网络隔离

多租户安全对于 Kubernetes 的企业级使用至关重要。多租户允许您让不同的团队使用同一个集群，同时防止未经授权访问彼此的环境。红帽 OpenShift 通过内核命名空间组合、SELinux、RBAC、Kubernetes（项目）命名空间和网络策略来支持多租户。

- **红帽 OpenShift 项目**是带有 SELinux 注释的 Kubernetes 命名空间。项目将跨团队、组和部门的应用隔离开来。本地角色和绑定用于控制谁能访问单个项目。

- **安全上下文约束**让您默认放弃特权，这非常重要，而且仍是最为有效的一种方式。红帽 OpenShift 的安全上下文约束（SCC）确保在默认情况下，OpenShift 工作节点上没有特权容器运行。默认拒绝访问主机网络和主机进程 ID。

在容器中部署基于微服务的现代化应用通常意味着要在多个节点上分布式部署多个容器。这些微服务需要相互发现和通信。出于网络防御方面的考量，您需要通过容器平台获得单个集群并对流量进行划分，以将该集群中的不同用户、团队、应用和环境隔离开来。您还需要工具来管理从外部访问集群以及从集群服务到外部组件的访问权限。实现网络隔离需要具备以下关键属性：

- **入口流量控制。**红帽 OpenShift 包含 CoreDNS，为容器集提供名称解析服务。红帽 OpenShift 路由器（HAProxy）支持入口和路由，来提供从外部对集群上运行服务的访问。两者都支持重新加密和直通策略：“reencrypt”在转发 HTTP 流量时对其进行解密和重新加密，而 “passthrough”则在不终止 TLS 的情况下通过流量。
- **网络命名空间。**网络防御的第一道防线来自网络命名空间。各个容器集合（称为“容器集”）都能获得自己所要绑定的 IP 和端口范围，因此能使节点上的容器集网络相互分隔开。容器集 IP 地址独立于红帽 OpenShift 节点所连接的物理网络。
- **网络策略：**红帽 OpenShift SDN 使用网络策略对容器集之间的通信进行精细的控制。在特定端口和特定方向，可控制网络流量从任何其他容器集至任何容器集。当网络策略配置在多租户模式下时，每个项目都会获得自己的虚拟网络 ID，从而在节点上将项目网络相互隔离。在多租户模式下（默认情况下），一个项目内的容器集可以相互通信，但来自不同命名空间的容器集不能向另一项目的容器集或服务发送数据包或从中接收数据包。
- **出口流量控制。**红帽 OpenShift 还提供利用路由器或防火墙的方法来控制在集群上运行的服务出口流量的能力。例如，您可以使用 IP 白名单来访问外部数据库。

11. 确保应用的访问安全

要确保应用安全，需要对应用用户加以管理并实施 API 身份验证和授权。

- **控制用户访问权限**

Web 单点登录（SSO）功能是现代化应用的一个关键组成部分。容器平台可以随附大量容器化服务，以供开发人员在构建应用时使用。红帽单点登录是受到全面支持的开箱即用型安全判定标记语言（SAML）2.0 或基于 OpenID Connect 的身份验证、Web 单点登录以及基于上游 Keycloak 项目的联合服务。红帽单点登录配备适用于红帽 Fuse 和红帽 JBoss 企业应用平台的客户端适配器。红帽单点登录可用于针对 Node.js 应用进行身份验证和 Web 单点登录，以及可与基于 LDAP 的目录服务整合，包括 Microsoft Active Directory 和红帽企业 Linux 身份管理。红帽单点登录还能与社交登录提供商集成，如 Facebook、Google 和 Twitter。

- **控制 API 访问权限**

对于由微服务构成的应用而言，API 是关键所在。这些应用拥有多个独立的 API 服务，会导致服务端点激增，所以需要使用额外的工具来加以监管。我们建议您使用 API 管理工具。红帽 3scale API 管理提供了多种多样的标准 API 身份验证和安全选项，这些选项可以单独或结合使用，以发放凭证和控制访问权限。

红帽 3scale API 管理中可用的访问控制功能要优于基本的安全和身份验证功能。您可以利用应用和帐户计划来限制对于特定端点、方法和服务的访问，并为用户组应用访问策略。通过应用计划，您可以为 API 的使用设置速率限值，并控制开发人员小组的流量。您可以按时段为传入的 API 调用设置限值，以便保护您的基础架构并使流量保持平稳。您还可以让已达到或超出速率限值的应用自动触发超额警报，并针对超出限值的应用定义相应的行为。

确保应用流量安全

本文第 10 章介绍了使用集群入口和出口选项来确保应用流量安全。对于基于微服务的应用，集群上服务之间的安全流量同样重要。服务网格可用于提供这个管理层。术语“服务网格”描述了在分布式微服务架构中构成应用的微服务网络以及这些微服务之间的交互。

红帽 OpenShift 服务网格基于开源 Istio 项目，在现有分布式应用上增加了一个透明层，用于管理服务间通信，而无需对服务代码进行任何修改。红帽 OpenShift 服务网格使用多租户操作器来管理控制平面生命周期，使得 OpenShift 服务网格能够在每个项目上使用。此外，OpenShift 服务网格不需要集群范围的 RBAC 资源。

红帽 OpenShift 服务网格提供发现、负载平衡、安全关键、服务间身份验证、加密、故障恢复、指标和监控。

3scale Istio Adapter 是一个可选的适配器，允许您标记在红帽 OpenShift 服务网格中运行的服务。

12. 可观测性

监控和审计红帽 OpenShift 集群的能力是保护集群及其用户免受不当使用的一个重要部分。红帽 OpenShift 包括内置的监控和审计，以及可选的日志堆栈。

OpenShift 容器平台服务连接由 Prometheus 及其生态系统组成的内置监控解决方案。可使用警报控制面板。集群管理员可以选择对用户定义项目启用监控。部署到红帽 OpenShift 的应用可以被配置用于充分利用集群监控组件。

审计事件是安全防护最佳实践，通常需要遵守监管框架。红帽 OpenShift 审计的核心采用云原生方法，从而同时提供集中化和弹性。红帽 OpenShift 默认在所有节点上启用主机审计和事件审计。红帽 OpenShift 为配置管理和访问审计数据提供非凡的灵活性。通过选择要使用的审计日志策略配置文件，您可以控制记录到 API 服务器审计日志的信息量。

监控、审计和记录数据受 RBAC 保护。项目数据可供项目管理员使用，集群数据可供集群管理员使用。

作为最佳实践，将集群配置为将所有审计和日志事件转发到安全信息和事件管理（SIEM）系统，来进行完整性管理、保留和分析。集群管理员可以部署集群日志记录，以汇总红帽 OpenShift 集群的所有日志，如主机和 API 审计日志，以及应用容器日志和基础架构日志。集群日志记录汇总整个集群节点的这些日志，并将其存储于默认的日志存储。有多种选项可用于将日志转发到您选择的 SIEM。

借助强大的生态系统扩展安全性

为了进一步增强容器和 Kubernetes 的安全性，或满足现有策略，您可以选择集成第三方安全防护工具。红帽广泛的认证合作伙伴生态系统提供解决方案，如：

- 特权访问管理。
- 外部证书颁发机构。
- 外部库和关键管理解决方案。
- 容器内容扫描程序和漏洞管理工具。
- 容器运行时分析工具。
- SIEM。

结论

部署基于容器的应用和微服务不仅仅是为了安全。您的容器平台需要提供一种既适用于开发人员、又适用于运维团队的体验。您需要一个安全至上、基于容器的企业级应用平台，这个平台不但要能够支持开发人员和运维人员，而且不能影响两个团队各自所需的功能，另外还要能够提升运维效率和基础架构利用率。

红帽 OpenShift 以可移植的标准 Linux 容器为核心构建而成，提供了多种内置安全功能，其中包括：

- 集成式构建和 CI/CD 工具，可确保 DevOps 实践的安全性。
- 经过增强的企业就绪型 Kubernetes，内置平台配置、合规性和生命周期管理。
- 能与企业身份验证系统相整合的强大 RBAC。
- 集群入口和出口的管理选项。
- 集成 SDN 和服务网格，支持网络微分段。
- 支持确保远程存储卷的安全。
- 红帽企业 Linux CoreOS 已经过优化，可大规模运行容器，具备强大隔离性。
- 实现运行时安全自动化的部署策略。
- 集成式监控、审计和日志记录。

红帽 OpenShift 还支持多种编程语言、框架和服务（图 4）。用于 Kubernetes 的红帽高级集群管理提供紧密整合的多集群管理。

红帽 OpenShift 可在 OpenStack、VMware、裸机、AWS、Google Cloud Platform（GCP）、Azure、IBM Cloud 以及支持红帽企业 Linux 的任意平台上运行。红帽还以公共云服务的形式，在 AWS 和 GCP 上提供红帽 OpenShift 专业版。Azure 红帽 OpenShift 由红帽和微软联合提供。AWS 上的红帽 OpenShift 服务由红帽和亚马逊联合提供。

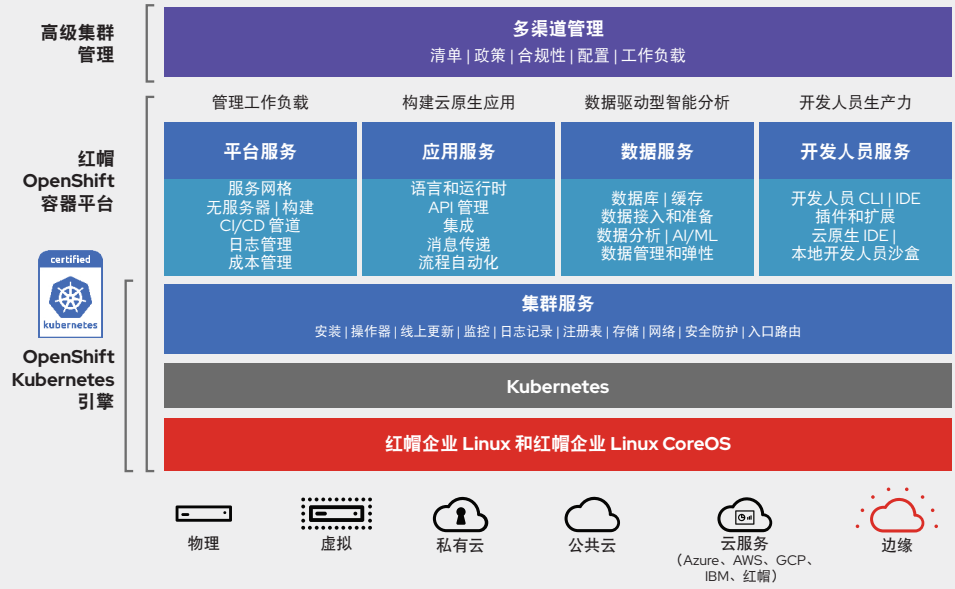

图 4. 红帽 OpenShift 提供跨物理虚拟、私有云和公共云的全功能技术堆栈。

二十多年来，红帽在向企业客户提供可信的开源解决方案方面，一直占据着领导地位。我们借助红帽 OpenShift 容器平台、用于 Kubernetes 的红帽高级集群管理等解决方案以及基于容器的红帽产品组合，来提供同样安全可信的容器。

关于红帽

红帽是世界领先的企业开源软件解决方案供应商，依托强大的社区支持，为客户提供稳定可靠而且高性能的 Linux、混合云、容器和 Kubernetes 技术。红帽帮助客户集成现有和新的 IT 应用，开发云原生应用，在业界领先的操作系统上开展标准化作业，并实现复杂环境的自动化、安全防护和管理。凭借一流的支持、培训和咨询服务，红帽成为《财富》500 强公司备受信赖的顾问。作为众多云提供商、系统集成商、应用供应商、客户和开源社区的战略合作伙伴，红帽致力于帮助企业做好准备，拥抱数字化未来。

销售及技术支持

800 810 2100
400 890 2100

红帽北京办公地址

北京市朝阳区东大桥路 9 号侨福芳草地大厦 A 座 8 层　邮编: 100020
8610 6533 9300

红帽官方微博　红帽官方微信

cn.redhat.com
#F26463_1220

版权所有 © 2020 Red Hat, Inc. 红帽、红帽 Logo、OpenShift 和 JBoss 是红帽在美国和其他国家/地区的商标或注册商标。Linux® 是 Linus Torvalds 在美国和其他国家/地区的注册商标。